OCL Computing Curriculum: Long Term Plan

Year 7

Brief overview
In year 7 students will arrive with varying levels of computing knowledge from the primary school they have attended and the technology they have access to at home. In order to build their confidence students will start to use a variety of different software and web apps to improve their digital literacy starting from logging on in lesson 1. Autumn 1 will give students opportunity to familiarize themselves with the computing lab and/or iPad. Many students will soon be creating social media accounts if they have not already. Student will be looking at what respectful online communication looks like and how messages online can seen without context. Students will then look at cyberbullying and create a presentation on the subject using MS PowerPoint. This allows students to cover important e-safety topics while still getting hands on experience with their computer and commonly used software. In Autumn 2 students will using another common software package MS Excel to work with data. The purpose of this unit is for students to understand how data can be collected, analyzed, and used. Spring 1 introduces students to networks and how they aid communication between computers. This unit will first look at the benefits of computer networks, how they are created and how they operate. The second half of the unit will focus on the Internet, smart devices and how this is changing the way we live our lives. Spring 2 will be the first experience of computer programming for many students. They will be using Scratch a web based visual programming language that allows students to create programs by connecting premade code blocks. Student will be able to explore the programming concepts of sequence, selection and iteration without worrying about syntax errors. This unit will set the foundation of students programming knowledge embedding key concepts that they will continue to explore throughout KS3 and KS4. Summer 1 will focus on creating text and image media while also looking at legal issues such as copyright law and plagiarism. This unit will also look at credibility, sourcing information and fake news. Finally in summer 2 students will look at how computers work. Students will first study hardware and the role of different components. Then they will look at software in general terms, instead of focusing on specific packages they will understand the role of different archetypes of software and operating systems. **Software Packages:** Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition throughout year 7 they will use MS Office(PowerPoint, Excel and Word), Scratch.mit.edu, and Photopea.com.

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	**Digital Literacy and E-Safety 1**	**Digital Literacy and E-Safety 2**	**Computer Networks**	**Python Unit 1 Python Basics**	**Multimedia 1 – Digital images**	**Computer systems**
Relevant core concepts	• Use a wide range of software and technology. • Display fundamental ICT Skills. • Create and edit a variety of media. • Be aware of the risks of technology and how they can be minimised. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT Skills. • Create and edit a variety of media. • Be aware of the risks of technology and how they can be minimised. • Maximise the use of Horizons.	• Understand what networks are and how they are used. • Recognise and predict technology trends • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT Skills. • Create and edit a variety of media. • Be aware of the risks of technology and how they can be minimised. • Recognise and predict technology trends. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT Skills. • Recognise computer hardware and understand how each component works. • Maximise the use of Horizons.

Relevant end points	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to create presentations and deliver them to an audience. • Students will be aware of the common risks of technology involving cyberbullying and how to avoid or report instances of each. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to create presentations and deliver them to an audience. • Students will be aware of the common risks of technology involving cyberbullying and how to avoid or report instances of each. • Students will be able to effectively use their Horizons device in a variety of contexts. •	• Students will be able to recognise the devices that make up computer networks. • Students will be able to predict future changes in technology with their understanding of computer networks history and fundamentals of how computers communicate. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to understand and create scratch code using block based coding. • Students will be able to develop a solution to a problem using computational thinking. • Students will be able to use Boolean logic (AND, OR, NOT) to make complex decisions in their programs. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to create and format documents in word processing software. • Students will understand the licensing rules on online media and how to find images under the creative commons licence. • Students will be able to recognise common features of written media such as blog posts and articles. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to recognise different computer hardware components and understand their role within the overall system. • Students will be able to effectively use their Horizons device in a variety of contexts.

Core declarative knowledge	• Usernames and passwords to log on
	• Features of common applications
	• Understanding which application is best for a given task.
	• How to navigate the computer
	• How to effectively get information from the internet
		• Cyberbullying is any form of bullying that uses a digital device.
		• Comments and messages should be respectful
		• Images used should be appropriate
		• Digital messaging should be clear and media used should be appropriate
		• Branding should be consistent across media products
			• What a computer network is.
			• The hardware that supports computer networks.
			  ○ Hub
			  ○ Router
			  ○ Server
			  ○ Network Cable
			• Understanding the different protocols and technologies that allow devices to connect.
			• What the internet is.
			• Services that run on the internet
			  ○ WWW
			  ○ Email
			  ○ IoT
				• Variables.
				• Sequence.
				• Input and Output
				• Storing and accessing data
					• Bitmap graphics.
					• Vector graphics.
					• Graphics properties.
					• Graphic file types
						• Input/output devices.
						• What is inside a computer.
						• General purpose systems.
						• Memory and storage types.
						• Application software.
						• System software.

			○ VoIP • The history and continued growth of computer networks.			
Core procedural knowledge	• Logging onto the school network • How to write an email • How to find and install apps on the iPad. • Using MS Office applications and choosing the correct application for a given task	• How cyberbullying can be stopped and reported. • How to create a PowerPoint presentation. • How to us text formatting functions in windows apps. • Finding and using appropriate images. • How to present to peers.	• Analysing how computers have aided communication around the world. • Analysing how each piece of hardware enhances a network. • Analysing the best network type for a given scenario. • Analysing the strengths and weaknesses of different methods of connection. • Understanding the different internet services and what they do. • Describe what the internet is and what is used for. • Explain the need for IP addresses. • Explain how data is broken up into packets to be sent across networks.	• Using built in subprograms • Output messages to the console • Get input from the keyboard • Store and retrieve data from variables. • Design and create programs that execute in sequence.	• Create and edit graphics. • Save and export images in the correct format. • Create media products to satisfy a client brief • Create pre production planning documents	• Categorise peripherals by their purpose. • Describe internal hardware and their purpose. • Understand and define different types of memory. • Compare storage types and select the most appropriate type for a scenario. • Design a computer for a given scenario. • Define application and system software.

Year 8

Brief overview
In year 8 students will continue to develop their digital literacy with multiple opportunities to have hands of experience with a variety of software packages and web applications. Students will build upon their practical skills from year 7 particularly in programming and media creation. Students will also begin to look at how computers represent data and solve problems. Autumn 1 will give students the opportunity to look at creating some more complex programs in Scratch. After a recap of selection and iteration student will start to look at how subroutines help us to organize and create one aspect of a program at a time, how instructions can be passed to other parts of a program and pull all this together to create a complex program. Autumn 2 will students will study the binary number system, why it is used by computers and how to convert between this and our denary number system. Once they have an understanding of binary numbers they will look at the logic gates AND, OR and NOT and their respective truth tables. In spring 1 students will look at using computational thinking to model and solve problems. They will also explore different methods of representing algorithms including using flowcharts and written pseudocode. Spring 2 students will move onto their second programming language Python. This first unit will focus again on the core programming concepts of Sequence, Selection and Iteration. As a text based language students will now encounter syntax errors and will need to interpret error messages to help debug their own code. In Summer 1 students will again have the opportunity to create and edit media. In this unit they will look at images in more detail studying both bitmap and vector graphics. Students will then look at using these image files to create keyframe animations. Finally in summer 2 students will have the opportunity to look at how websites are created and create their own website. While creating web pages students will also focus on good web design practiced and the importance of making websites accessible to everybody. Software Packages: Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition they will use scratch.mit.edu, logic.ly, Python, MU IDE, photopea.com, wickeditor.com, Notepad ++

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	**Scratch Programming 2**	**Binary and Boolean Logic**	**Computational Thinking and Algorithms**	**Programming with Python 1**	**Multimedia 2**	**Developing for the Web**
Relevant core concepts	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT skills. • Use computational thinking skills to solve real world problems. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Create and edit a variety of media. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Create and edit a variety of media. • Be aware of the risks of technology and how they can be minimised. • Understand what networks are and how they can be used. • Recognise and predict technology trends. • Maximise the use of Horizons.
Relevant end points	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to understand and create scratch code using block based coding. • Students will be able to develop a solution to a problem using computational thinking. • Students will be able to use Boolean logic (AND, OR,	• Students will be able to recognise the logic gates (AND, OR and NOT). • Students will be able to complete truth tables for each logic gate and simple circuits. • Students will be able to understand logic scenarios and draw the logic circuit that represents them. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to break down problems into manageable chunks and use algorithmic thinking to create solutions for them. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to create Python programs that utilise user input. • Students will be able to debug Python programs using IDE error reporting tools. • Students will be able to effectively use their	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will understand the difference between bitmap and vector graphics. • Students will recognise different styles and techniques used in animations. • Students will be able to create animations.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to create and edit webpages using HTML and CSS. • Students will understand the importance of accessibility on the web and how to make web pages accessible to everyone. • Students will understand how to link and move between different parts and to different media in a website.

	NOT) to make complex decisions in their programs. • Students will be able to effectively use their Horizons device in a variety of contexts.			Horizons device in a variety of contexts.	• Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will be able to effectively use their Horizons device in a variety of contexts.

Core declarative knowledge	• Variables.
	• Subroutines.
	• Sequence.
	• Selection (if).
	• Iteration.
	  ○ Count controlled (for)
	  ○ Condition controlled (While)
	• Operators
	  ○ Arithmetic operators (=, <, >)
	  ○ Logical operators(and, or, not)
		• Binary numbers.
		• Binary Place Value Tables
		• Logic gates
		  ○ AND
		  ○ OR
		  ○ NOT
		• Logic circuits
		• Truth tables.
			• Computational thinking.
			  ○ Abstraction.
			  ○ Decomposition.
			  ○ Pattern recognition.
			  ○ Algorithmic thinking.
			• Designing and creating algorithms.
			  ○ Flowcharts.
			  ○ Pseudocode.
			• Searching and sorting algorithms.
				• Variables.
				• Syntax (Python).
				• Debugging.
				  ○ Syntax errors
				  ○ Logic errors
				• Sequence.
				• Selection (if).
				• Iteration.
				  ○ Condition controlled (While)
					• Bitmap graphics.
					• Vector graphics.
					• Graphics properties.
					• Animation techniques
					  ○ Frame by frame.
					  ○ Tweening.
					  ○ Onion skinning.
					• Animation types
					  ○ Cell.
					  ○ Cut-out.
					  ○ Stop motion.
					  ○ Claymation.
					  ○ Flip-book.
					• Purposes of animation
					  ○ Advertise
					  ○ Educate
					  ○ Entertain
					• Storyboard features
						• HTML.
						• CSS.
						• Accessibility.
						• Navigation and hyperlinks.

Core procedural knowledge	• Store and retrieve data from variables. • Design and create programs that execute in sequence. • Design and create programs that use selection to make decisions. • Combine operators to make more complex decisions. • Design and create programs that use iteration to repeat code. • Design and create subroutines that can be called by the main program as needed.	• Represent denary numbers in binary. • Covert between binary and denary. • Select the correct logic gate for a given scenario • Construct truth tables for logic gates (AND, OR, NOT). • Construct truth tables for simple logic circuits. • Create, modify and interpret simple logic diagrams.	• Understand different types of searching and sorting algorithms. • Compare algorithms by best and worst case scenarios. • Apply computational thinking techniques to real world problems. • Represent algorithms in Pseudocode and using flowcharts.	• Store and retrieve data from variables. • Design and create programs that execute in sequence. • Design and create programs that use selection to make decisions. • Design and create programs that use iteration to repeat code. • Debug existing programs that do not execute correctly using IDE error reports.	• Create and edit graphics. • Save and export images in the correct format. • Create images of specific dimensions and quality. • Use various animation techniques to create a short animation. • Create mind maps with nodes and sub nodes • Plan an animation using a storyboard	• Create webpages using HTML. • Style webpages using CSS. • Use the correct tags to add multimedia to a website. • Make pages accessible to as many people as possible. • Navigate between pages in a website by creating hyperlinks. • Create a multipage about me website.

Year 9

Brief overview
In year 9 students will expand upon all of the skills they have learnt so far with 3 units focused on creating programs. Students will also look at 2 of the most popular career paths in Computer Science Cybersecurity and Data Science. These units will expand upon previous security and data units with a focus on real world examples of cyber threats and handling data. Autumn 1 starts with students looking at binary again this time with a focus on how it is used to represent data. Students will gain insight into how computers actually store the files on the computer and how binary can interpreted to display images or play audio. Autumn 2 delves deeper into the Python programming language focusing again on the core programming concepts alongside data types (integer, float, string and Boolean) and using the Random library to generate random numbers. Spring 1 gives students the opportunity to put everything they have learned in the previous programing units to plan, create and test a complex program. The unit will start at looking how subroutines are used in Python then students will be given a scenario that they will plan, create and evaluate a program for. Spring 2 will introduce students to cybersecurity, students will look at different cyber attacks and system vulnerabilities before looking at how software and networks are can be designed to protect against these attacks. In summer 1 students will be looking at mobile app design, this requires a different approach to programming than they have used previously. Students will create a Graphical User Interface and add functionality driven by user interaction. Finally students will finish KS3 by looking at data science and big data. This unit focusses on different ways to present data using visualizations before looking at how and why large data sets needs to be investigated and cleaned before they can be used. **Software Packages:** Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition they will use Python, Mu IDE, App Lab (code.org), MS Excel, MS Access

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	Binary and Data Representation	Programming with Python 2	Computer Systems 2	Cybersecurity	Programming With Python 3	Data Science
Relevant core concepts	• Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT Skills. • Recognise computer hardware and understand how each component works. • Maximise the use of Horizons.	• Display fundamental ICT skills. • Be aware of the risks of technology and how they can be minimised. • Understand what networks are and how they are used. • Recognise and predict technology trends. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT skills. • Be aware of the risks of technology and how they can be minimised. • Use computational thinking skills to solve real world problems. • Recognise and predict technology trends. • Maximising the use of Horizons.
Relevant end points	• Students will understand how binary numbers can be used to represent data stored in computers. • Students will be able to convert between binary and denary numbers. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to create Python programs that utilise user input. • Students will be able to create complex python programs that use selection and iteration. • Students will be able to debug Python programs using IDE error reporting tools. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will understand the purpose of the CPU. • Students will understand the different types of memory and their role in the computer. • Students will be able to compare storage types for a given scenario. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will be able to select and use the correct device for a given task. • Students will be aware of the common features of online scams. • Students will recognise the main types of malware and how they affect a computer system. • Students will understand the steps that can be taken to protect computers and data from hacking and malware.	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to create Python programs that utilise user input. • Students will be able to create complex python programs that use selection and iteration. • Students will be able to debug Python programs using IDE error reporting tools. • Students will be able to effectively use their	• Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will understand how organisations use the data they gather. • Students will understand the unique challenges that come with large data sets. • Students will be able to interrogate data sets and create visualisations from them. • Students will be able to effectively use their Horizons device in a variety of contexts.

				• Students will be able to effectively use their Horizons device in a variety of contexts.	Horizons device in a variety of contexts.	

Core declarative knowledge	• Binary numbers.
	• Character sets.
	• Representing images in binary.
		• Variables.
		• Data types
		  ○ Int.
		  ○ String.
		  ○ Float.
		  ○ Boolean.
		• Syntax (Python).
		• Importing modules.
		• Debugging.
		  ○ Syntax errors
		  ○ Logic errors
		• Sequence.
		• Selection (if).
		• Iteration.
		  ○ Count controlled (for)
		  ○ Condition controlled (While)
			• The function of the CPU.
			• Von Neumann architecture.
			• FDE cycle.
			• Memory and storage types.
				• Data protection.
				• Social engineering.
				• Hacking.
				• Malware.
				• Creating secure systems.
					• Variables.
					• Data types
					  ○ Int.
					  ○ String.
					  ○ Float.
					  ○ Boolean.
					• Syntax (Python).
					• Importing modules.
					• Debugging.
					  ○ Syntax errors
					  ○ Logic errors
					• Sequence.
					• Selection (if).
					• Iteration.
					  ○ Condition controlled (While)
						• Data vs information.
						• Analysing data.
						• Visualising data.
						• **PPDAC** cycle
						• Why data is useful.
						• Answering real-world problems using data

Core procedural knowledge	• Represent denary numbers in binary. • Covert between binary and denary. • Add binary numbers together. • Use character sets to represent letters and characters in binary. • Understand how images are stored in binary.	• Store and retrieve data from variables. • Cast data to different data types as required. • Design and create programs that execute in sequence. • Design and create programs that use selection to make decisions. • Design and create programs that use iteration to repeat code. • Debug existing programs that do not execute correctly using IDE error reports. • Import modules to use pre written functions.	• Understand and define different types of memory. • Compare storage types and select the most appropriate type for a scenario. • Design a computer for a given scenario.	• Understand why data is desired by hackers. • Understand data protection legislation. • Understand how humans can be the weak point in a system. • Recognise common social engineering scams. • Define different methods of hacking. • Define different types of Malware. • Understand how software and networks can be created with defence in mind.	• Store and retrieve data from variables. • Cast data to different data types as required. • Design and create programs that execute in sequence. • Design and create programs that use selection to make decisions. • Design and create programs that use iteration to repeat code. • Debug existing programs that do not execute correctly using IDE error reports. • Import modules to use pre written functions • How to refactor code to improve it. • How to build a larger program over multiple sessions.	• Understand why data is useful to organisations. • Understand the challenges that come with large data sets. • Interrogate data to get specific information. • Analyse data to spot trends. • Create visualisations from data to make it easier to understand and identify patterns. • Plan questions to gather useful data

Year 10

Brief overview
In year 10 assuming 3 lessons per week students will start with 2 programming lessons using Python (topic 6) and 1 theory lesson focusing on topics 1, 2 and 3 primarily. From summer 1 when most programming topics have been covered this will swap to 1 programming lesson and 2 theory lessons. The programming lessons although focusing mainly on problem solving with programming and paper 2 topics will also cover and allow application of algorithms (topic 1) and characteristics of programming languages (Topic 3). Students will start learning how programming languages work and how to create computer programs right from the start of Autumn1 and continue this throughout the year. They will use the Python programming language to complete the practical aspects of their lessons. The introduction to the language will assume that students have not used Python or had very limited experience with it. Autumn 1 will mainly focus input, output and using variables, students will not use selection or iteration techniques until Autumn 2. Due to the amount of different programming techniques that students need to master throughout the GCSE the complexity of the programs they create will build much quicker than KS3. By the end of year 10 students should have encountered and have a good familiarity with every programming technique required for GCSE Computer Science and have applied them to a complex programming project. The more theory based lessons will focus on Computer Systems, Boolean Logic, Algorithms, Cybersecurity and Data Representation. **Software Packages:** Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition they will use Python, Mu IDE

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	Programming: Sequence, variables, input and output. Theory: Computer Systems.	Programming: Selection, Iteration and validation. Theory: Boolean Logic.	Programming: Subroutines and structured programming. Theory: Computational thinking and Algorithms.	Programming: Working with strings, arrays and character sets. Theory: Cybersecurity.	Programming: Working with external files. Theory: Data representation (number systems).	Programming: Python Turtle Theory: Data representation (Images and audio).
Relevant core concepts	• Use a wide range of software and technology. • Display fundamental ICT skills. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Recognise computer hardware and understand how each component works. • Recognise and predict technology trends. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT skills. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Recognise computer hardware and understand how each component works. • Understand simple Boolean logic. • Recognise and predict technology trends. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Be aware of the risks of technology and how they can be minimised. • Recognise and predict technology trends. • Maximising the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Display fundamental ICT skills. • Modify and create computer programs. • Create and edit a variety of media. • Use computation thinking to solve real world problems. • Understand simple Boolean logic. • Maximise the use of Horizons.
Relevant end points	<u>Theory</u> • Students will recognise and describe Von Neumann architecture. • Students will be able to explain the Fetch Decode Execute cycle. • Students will recognise different memory types and understand the role of each type in computer systems. • Students will be able to recognise different storage types. <u>Programming</u>	<u>Theory</u> • Students will be able to compare storage types suitability for a given scenario. • Students will be able to design an appropriate computer system for a given scenario. • Students will be able to recognise the logic gates (AND, OR and NOT). • Students will be able to complete truth tables for each logic gate and simple circuits.	<u>Theory</u> • Students will be able to use computational thinking techniques to build algorithms in pseudocode and flowcharts. • Students will be able to trace algorithms to understand what is happening at each point and locate logic errors. <u>Programming</u> • Students will be able to translate pseudo code into Python code. • Students will be able to create procedures and functions.	<u>Theory</u> • Students will be aware of the common features of online scams. • Students will recognise the main types of malware and how they affect a computer system. • Students will understand the steps that can be taken to protect computers and data from hacking and malware.	<u>Theory</u> • Students will understand how binary numbers can be used to represent data stored in computers. • Students will be able to convert between binary and denary numbers. • Students will be able to add, subtract and multiply binary numbers. • Students will be able to convert between	<u>Theory</u> • Students will be able to calculate the size of files. • Students will understand how computers represent images and sound as well as the properties that affect the quality of these files. • Students will understand the benefits and drawbacks of lossy and lossless compression techniques. <u>Programming</u>

	• Students will be able to explain the difference between different translators. • Students will be ablet o write programs that execute code in sequence using input and output. • Students will be able to debug code using IDE tools. • Students will be able to read and produce flowcharts. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will be able to understand logic scenarios and draw the logic circuit that represents them. Programming • Students will be able to import code from modules. • Students will be able to create programs that use selection and iteration. • Students will be able to use trace tables to find logic errors in their code. • Students will be able to validate data that is input into their programs. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will understand the scope of variables. • Students will be able to plan and create a larger program using a structured approach. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will understand how computers and networks can be designed with security in mind. Programming • Students will be able to use methods to manipulate strings and lists. • Students will be able to work with data in lists and 2D lists. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	binary and hexadecimal numbers. Programming • Students will be able to load data from and save data to text and csv files. • Students will be able to use comments in their programs and understand good programming practices. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will understand how to import and use modules. • Students will understand how to draw graphics with Python turtle. All lessons • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.

Core declarative knowledge	Theory
	• The function of the CPU.
	• Von Neumann architecture.
	• FDE cycle.
	• Memory and storage types.
	• Application software.
	• System software.
	
	Programming
	• Variables
	• Input and output
	• Program sequence
	• Modules
	• Translators
	• Machine code
	• Data Types
	• Casting
		Theory
		• Binary numbers.
		• Logic gates
		  ○ AND
		  ○ OR
		  ○ NOT
		• Logic circuits
		• Truth tables.
		
		Programming
		• Variables
		• Input and output
		• Program sequence
		• Data Types
		• Casting
		• Debugging and exception handling
		• Importing libraries
			Theory
			• Computational thinking.
			  ○ Abstraction.
			  ○ Decomposition.
			  ○ Pattern recognition.
			  ○ Algorithmic thinking.
			• Designing and creating algorithms.
			  ○ Flowcharts.
			  ○ Pseudocode.
			
			Programming
			• Variables
			• Input and output
			• Program sequence
			• Data Types
			• Casting
				Theory
				• Data protection.
				• Social engineering.
				• Hacking.
				• Malware.
				• Creating secure systems.
				
				Programming
				• Variables
				• Input and output
				• Program sequence
				• Data Types
				• Casting
				• Debugging and exception handling
				• Importing libraries
					Theory
					• Binary numbers.
					• Conversion between binary numbers.
					• Binary addition.
					• Binary shift.
					• Signed binary integers.
					• Hexadecimal numbers.
					
					Programming
					• Variables
					• Input and output
					• Program sequence
					• Data Types
					• Casting
					• Debugging and exception handling
						Theory
						• Units of data storage.
						• How data is represented in binary
						  ○ Files
						  ○ Images
						  ○ Sounds
						• Lossy and lossless compression.
						• Calculating files sizes.
						
						Programming
						• Variables
						• Input and output
						• Program sequence
						• Data Types
						• Casting
						• Debugging and exception handling

	• Debugging and exception handling	• Arithmetic expressions (operators) • Logical expressions • Selection statements • Iteration (while and for) • Tracing Algorithms • Validating input	• Debugging and exception handling • Importing libraries • Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing	• Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods	• Importing libraries • Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods • Handling external files • Using comments	• Importing libraries • Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods • Using comments • Making graphics with Python Turtle
Core procedural knowledge	Theory • Understand and define different types of memory. • Compare storage types and select the most appropriate type for a scenario. • Design a computer for a given scenario. • Describe how each component of the CPU operates. Programming • Output to the console • Import code from a module • Use the input() function to get user input. • Store and retrieve data from variables. • Cast data to a different type (str to int etc) • Understand error messages and how to handle the errors that they represent.	Theory • Construct truth tables for logic gates (AND, OR, NOT). • Construct truth tables for simple logic circuits. • Create, modify and interpret simple logic diagrams. Programming • Use the random and time libraries to enhance programs. • Make decisions in programs with selection statements. • Use operators to calculate • Use Boolean logic to make more complex decisions. • Repeat code using for and while loops. • Using trace tables to find logic errors • Validate data with loops and try/except	Theory • Apply computational thinking techniques to real world problems. • Represent algorithms in Pseudocode and using flowcharts. Programming • Translate pseudocode to Python code. • Create subroutines. • Create subroutines that return a value. • Pass data to and from subroutines ensuring that all variables used are in scope. • Plan programs using a structured approach. • Create and use a test plan to test a programs robustness.	Theory • Understand why data is desired by hackers. • Understand data protection legislation. • Understand how humans can be the weak point in a system. • Recognise common social engineering scams. • Define different methods of hacking. • Define different types of Malware. • Understand how software and networks can be created with defence in mind. Programming • Access and manipulate data in strings. • Access and manipulate data in lists.	Theory • Represent denary numbers in binary. • Covert between binary and denary. • Add binary numbers together. • Use binary shift to multiply binary numbers. • Use character sets to represent letters and characters in binary. • Convert between binary and hexadecimal numbers. Programming • Read, write and update data in external files. • Use comments to improve the reusability of code.	Theory • Define the terms bit, byte, kilobyte, megabyte and gigabyte. • Convert positive binary numbers (0-255) into denary and vice versa. • Add 2 binary integers and explain binary overflow. • Understand the use of binary shifts. • Understand the use of binary codes to represent characters. • Understand the term character set. • Explain how the size of image and sound file sizes are affected by changes to resolution, sampling intervals and bit depth. Programming • Import a library and use subprograms from it to build a complex program • Implement mathematical functions in Python • Control movement in 2D space using Python and cartesian coordinates

Year 11

Brief overview
In year 11 assuming 3 lessons per week students will be continuing with the model of 1 practical programming lesson and 2 theory based lessons that they moved to towards the end of year 10. It is expected that by the end of Spring 1 students will have finished all or most of the content allowing time for focused revision throughout Spring 1 and Summer 1 in preparation for the exams. As they will have covered all the programming techniques that they need to solve problems with Python (Topic 6) in year 10 programming lessons will be used to recap and build mastery focusing in some of the techniques that students regularly struggle. The final 2 theory topic from Paper 1 (topic 4 and topic 5) will be the main focus of the theory lessons. Autumn 1 starts with some recapping of the most common programming techniques that are used in almost all Python programs. The first lesson back with cover some of the very basics as many students will not have created a program over the summer. The complexity of programs will build very quickly so that in Spring 1 they can focus on independently building solutions to a variety of problems similar to those they will encounter in the assessment. Theory lessons will start looking at some specific algorithms for searching and sorting. Students will be able to recognize and compare these algorithms allowing them to choose the best one for a given situation and justify their choice. In Autumn 2 they will look at computer networks and spring 1 will focus on the issues and impact of computing. This unit will cover some of the same material as the cybersecurity unit in year 10, however, now students will be focusing more on the impact of cybercrimes when they occur. **Software Packages:** Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition they will use Python, Mu IDE

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	Programming: Exam style question practice. Theory: Searching and sorting algorithms.	Programming: Exam style question practice. Theory: Computer networks	Programming: Exam style question practice. Theory: Moral, legal, ethical and environmental technology concerns.	Catch up and revision	Catch up and revision	
Relevant core concepts	• Use a wide range of software and technology. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Modify and create computer programs. • Understand what networks are and how they are used. • Use computational thinking skills to solve real world problems. • Maximise the use of Horizons.	• Use a wide range of software and technology. • Be aware of the risks of technology and how they can be minimised. • Modify and create computer programs. • Use computational thinking skills to solve real world problems. • Maximise the use of Horizons.			
Relevant end points	<u>Theory</u> • Students will be able to compare searching and sorting algorithms based on best and worst case scenarios. • Students will be able to search for a specific item using a specified search algorithm. • Students will be able to sort data using a specified sorting algorithm. <u>Programming</u> • Students will be able to create programs that use sequence, selection and iteration.	<u>Theory</u> • Students will be able to identify the type of network that is in use/suitable for a given scenario. • Students will recognise different network topologies. • Students will understand how data is moved across networks including across the internet. • Students will be aware of the protocols used in data transmission. <u>Programming</u> • Students will be able to import code from modules. • Students will be able to create programs that use selection and iteration.	<u>Theory</u> • Students will have an awareness of the ethical, cultural, legal and environmental impacts of digital technology. • Students will understand key legislation that affects digital technology and intellectual property use in the UK. • Students will be able to explain the impact of cybercrimes and their consequences. <u>Programming</u> • Students will be able to use methods to manipulate strings and lists. • Students will be able to work with data in lists and 2D lists.			

	• Students will be able to get user input for their programs. • Students will be able to debug errors in their programs. • Students will be able to use Python methods to manipulate lists and validate data. <u>All lessons</u> • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	• Students will be able to use trace tables to find logic errors in their code. • Students will be able to validate data that is input into their programs. <u>All lessons</u> • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.	<u>All lessons</u> • Students will have the resilience to adapt to new software that may be required in their further education or professional careers. • Students will be able to select and use the correct device for a given task. • Students will be able to effectively use their Horizons device in a variety of contexts.			

Core declarative knowledge	<u>Theory</u>
	• Searching algorithms
	  ○ Linear search
	  ○ Binary search
	• Sorting algorithms
	  ○ Bubble sort
	  ○ Merge sort
	• Comparing algorithms.
	
	<u>Programming</u>
	• Variables
	• Input and output
	• Program sequence
	• Machine code
	• Data Types
	• Casting
	• Debugging and exception handling
	• Importing libraries
	• Arithmetic expressions (operators)
		<u>Theory</u>
		• Network types
		  ○ LAN
		  ○ WAN
		• Network topologies
		  ○ Star
		  ○ Mesh
		  ○ Bus
		• Network protocols
		  ○ TCP
		  ○ IP
		  ○ HTTP
		  ○ Ethernet
		• The TCP/IP model
		• Network hardware
		• Media transmission
		• Peer-to-peer networks
		• The client-server model
		• Network maps
		
		<u>Programming</u>
			<u>Theory</u>
			• Impacts of technology on wider society:
			  ○ Ethical issues
			  ○ Legal issues
			  ○ Cultural issues
			  ○ Environmental issues
			  ○ Privacy issues
			• Legislation relevant to computer science:
			  ○ The Data Protection Act 2018
			  ○ Computer Misuse Act 1990
			  ○ Copyright Designs and Patents Act 1988
			  ○ Software licences
			
			<u>Programming</u>
			• Variables
			• Input and output			

	• Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods • Handling external files • Using comments	• Variables • Input and output • Program sequence • Machine code • Data Types • Casting • Debugging and exception handling • Importing libraries • Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods • Handling external files • Using comments	• Program sequence • Machine code • Data Types • Casting • Debugging and exception handling • Importing libraries • Arithmetic expressions (operators) • Selection statements • Iteration (while and for) • Validating input • Subroutines (procedures and functions) • Scope • Constants • Testing • Arrays and lists • List and string methods • Handling external files • Using comments			

Core procedural knowledge	<u>Theory</u> • Locate items in data sets using searching algorithms. • Sort data sets using sorting algorithms. • Compare algorithms based on best and worst case scenarios. • Create searching and sorting algorithms in Python. <u>Programming</u> • Use the input() function to get user input. • Store and retrieve data from variables. • Cast data to a different type (str to int etc) • Understand error messages and how to handle the errors that they represent. • Use the random and time libraries to enhance programs. • Make decisions in programs with selection statements. • Use Boolean logic to make more complex decisions. • Repeat code using for and while loops. • Validate data with loops and try/except • Create subroutines. • Create subroutines that return a value. • Pass data to and from subroutines ensuring that all variables used are in scope. • Plan programs using a structured approach. • Create and use a test plan to test a programs robustness.	<u>Theory</u> • Analysing different topologies effectiveness for a given scenario. • Recognising which networks are categorised as LAN or WAN. • Describing the role of network protocols. • Describe the layers of the TCP/IP model. • Describe how data is broken into packets to be sent across networks. • Describe how data is routed across networks • Describe how different network hardware contributes to the overall system <u>Programming</u> • Access and manipulate data in strings. • Access and manipulate data in lists. • Read, write and update data in external files. • Use comments to improve the reusability of code.	<u>Theory</u> • List ethical, cultural and environmental issues for a given scenario. • List items of legislation that relate to digital technology. • Discuss the impact of digital technology on wider society. • Discuss the impact of manufacture, disposal, upgrading and replacing digital technology. • Discuss the impact of e-waste. • Describe legislation relevant to computing. <u>Programming</u> • Use the input() function to get user input. • Store and retrieve data from variables. • Cast data to a different type (str to int etc) • Understand error messages and how to handle the errors that they represent. • Use the random and time libraries to enhance programs. • Make decisions in programs with selection statements. • Use Boolean logic to make more complex decisions. • Repeat code using for and while loops. • Validate data with loops and try/except • Create subroutines. • Create subroutines that return a value. • Pass data to and from subroutines ensuring that all variables used are in scope. • Plan programs using a structured approach. • Create and use a test plan to test a programs robustness. • Access and manipulate data in strings.			

			• Access and manipulate data in lists. • Read, write and update data in external files. • Use comments to improve the reusability of code.			